# Research Review

There are three developments that made a big impact on AI planning. The problem solving program STRIPS, the planning graph analysis approach called GRAPH PLAN, and the testing of satisfiability instead of deduction. These developments have contributed greatly to AI planing and AI as whole. Difficult problems such as self-driving cars depend heavily on approaches inspired by these developments.

## STRIPS

STRIPS stands for STandford Research Institute Problem Solver[1]. It takes a "world model" with inital conditions and applies operators that change the state of the world model. It continues to do so until the conditions satisfy that of the goal world model. STRIPS was developed to solve a problem in robotics where robots have a large number of conditions that describe a world model. This problem solver uses first-order calculus to describe the world model. For example if a box *a* is at location A and box *b* is at location B, then the state would be described as:

$$at(a, A), at(b, B)$$

$$(\forall o \forall x \forall y) at(o, x) \wedge (x \neq y) => \sim at(o, y)$$

which states that a is at Location A and b is at location B. Also that for all elements o and locations x and y. If o is at x then it is not at y.

Operators describe an action that will consequently change the state of the world model. For example

$$move(o, x)$$

signifies an action of moving a box *o* to location *x*.  When a goal state is described. the program can simple seach for operators that get from the inital state to the goal state.

The search strategy for solving STRIPS could simply be applying operators to create new world models until the goal world model is created. A more efficent solution would be to find the difference between the initial or current model and the goal and determine which operators can reduce the differences.

STRIPS representation of operaters performing actions have been influential to all planning systems[4]. Including graph analysis systems like GRAPHPLAN.

## GRAPHPLAN

GRAPHPLAN is an algorithm based on a paradigm called Planning Graph Analysis[2]. The approach is to build a constraint *graph* based on STRIPS-like domains, that describes the operators (actions) and the conditions (literals) before and after and action is made. This graph allows for interleaving of conditions where one action will undo one condition, making another action impossible to perform. It also will show which actions do not change a condition allowing an algorithm to use or ignore the action in order to find the shortest path to the goal model.

GRAPHPLAN usings this Graph Analysis approach to find an optimal solution or determine that there's no solution at all. It does this by starting at the inital state with all the initial conditions. Chooses an action and checks the Planning Graph to see if the resulting conditions validate the goal state. If not, it continues to apply another action. More specifically, GRAPHPLAN uses a backward-chaining search strategy [2-3.2] that looks at the previous actions that add conditions needed to satisfy the current state. Recursively doing this allows the algorithm to try different (exlusive) actions until the right conditions are met. If not the algorithm will fail with no solution.

GRAPHPLAN performs well and is guaranteed to find the shortest path to the goal world model. Applied with heurisitics, it has out performed other partial-order planners (planners that are non-linear meaning the order of certain operations do not matter). However, it has it's drawbacks as well. GRAPHPLAN is a deductive planning approach, meaning it takes prior known state (could be goal or initial) to determine the next state. This doesn't scale well due to the large number of possible "resulting" states that could be determined by the prior know state. Another approach was developed to look at constraint graphs and solve the problem as a Constraint Statisfaction Problem.

# SATPLAN

SATPLAN or Planning as Satisfiability is a planning approach that, instead of deducing that an initial state and become a goal state through a set of actions or not at all, defines constraints on the actions, initial state and goal state and lets the algorithm satisify the constraints. It does this by searching for actions that when applied, is then checked if the constraints are satisfied with a result of either TRUE or FALSE. This guarentees a valid plan will result if possible. SATPLAN has the advantages that it can be very flexible dealing with conditions that must be satisfied at "*any intermediate state*" (Selman) [3-4].

Combined with greedy local search algorithms, SATPLAN can solve large constraint-based planning systems in minutes. This has led to a new way of thinking about planning problems in terms of constraints. SATPLAN has even been combined with GRAPHPLAN to convert STRIPS problems to Boolean satisfiable problems. This algorithm is called the BLACKBOX planner[5].

---

**Research Papers**

1. Richard E. Fikes, Nils J. Nilsson. "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving", (Winter 1971) LINK
2. A. Blum and M. Furst, "Fast Planning Through Planning Graph Analysis", Artificial Intelligence, 90:281--300 (1997) LINK

- [3.2] Searching for a plan

3. H. A. Kautz and B. Selman, “Planning as satisfiability”. In Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI'92), pages 359-363, (1992) [LINK] (http://lia.deis.unibo.it/Courses/AI/applicationsAI2009-2010/articoli/Planning_as_Satisfiability.pdf)
   - [3-4] Planning as Satisfiability
4. Artificial Intelligence: A Modern Approach by Norvig and Russell LINK